

Simulate with IVerilog & View the waveform with gtkwave

- Written by: Khanh N. Dang (khanh [at] u-aizu [dot] ac [dot] jp)
- Last update: Dec 20, 2024

The simulation script is available at the `hw` folder. The following are some explanations.

Icarus Verilog (IVerilog)

Icarus Verilog is a Verilog simulation and synthesis tool. It operates as a compiler, like gcc. It compiles source code written in Verilog (IEEE-1364) into some target format.

How to run

Like other compilers, the sequence of compiling files should be bottom-up. For example, there are two source files, `one.v` and `two.v`. File `two.v` calls file `one.v`. The way to compile these two files:

```
$ iverilog one.v two.v
```

For this command, like gcc, `a.out` is created under binary format. In order to execute it, **IVerilog** uses `vvp` command.

```
$ vvp a.out
```

To specify another name for the binary, please add `-o` option like in gcc:

```
$ iverilog -o $(your_name) one.v two.v
```

Like in gcc, **IVerilog** also supports `-Wall` option to report warnings/errors in detail.

```
$ iverilog -Wall -o $(your_name) one.v two.v
```

GTKWave

GTKWave is a fully featured **GTK++** based wave viewer for UNIX, Win32, and MAC OS which reads LXT, LXT2, VZT, FST, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing.

How to run In order to obtain waveform, the following code should be included in `testbench` file.

```
module Test;
    ...
    initial begin
        $dumpfile("$(your_name).vcd");
    end
endmodule
```

```

        $dumpvars(0, $(your_module_name));
    end
    ...
endmodule

```

Side Note: In order to extract the waveform of arrays in design, the following code is an example.

For example, your design has an array of registers.

```

module Mem (...);
    ...
    reg [M:0] data [N:0]
    ...
endmodule

```

Your **testbench** should include:

```

module Test;
    integer i;
    ...
    initial begin
        $dumpfile("$(your_name).vcd");
        $dumpvars(0, $(your_module_name));
        for (i = 0; i < M; i = i + 1)
            $dumpvars(1, full.path.to.array.data[i]);
    end
    ...
endmodule

```

Next, a **vcd** file is created after compiling the RTL code. Using GTKWave to observe the waveform.

```
$ gtkwave $(your_name).vcd
```

For quick reload configurations in simulating/debugging phase, please save your modified configuration of waveform into **.gtkw** file. For GUI, click **File -> Write Save File As**. Then:

```
$ gtkwave $(your_name).vcd $(your_config).gtkw
```